



SAND REPORT

SAND2001-2560

Unlimited Release

Printed September 2001

SIERRA Requirements Management Process Overview

H. C. Edwards, Pam W. Harris, and Shelley M. Eaton

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/ordering.htm>



SIERRA Requirements Management Process Overview

H. Carter Edwards, Pam W. Harris, and Shelley M. Eaton
Production Computing / SIERRA Architecture Department
Engineering Sciences Center
Sandia National Laboratories
Box 5800
Albuquerque, NM 87195-0827

Abstract

The objective of requirements management for SIERRA is to establish a disciplined process whereby managers and developers of the SIERRA infrastructure and the SIERRA applications share a common understanding of both the requirements specifications and how to evaluate, approve, and communicate requirement changes. This document presents a high-level description of the SIERRA Requirements Management Process. This process is critical to SIERRA's ability to produce the right products on schedule, within budget, and with high quality.

The purpose of the SIERRA Requirements Management Process is to standardize the manner in which each SIERRA software project manages its requirements during the life span of a software project. A disciplined requirements management process ensures all requirement changes are consistently documented, objectively evaluated, agreed upon, prioritized, and tracked to completion. A disciplined process improves quality, customer satisfaction, and productivity, while decreasing rework, costs, and risk. Each SIERRA project may extend the Requirements Management Process presented in this document to meet the project's specific needs. The project-specific extensions will be documented as appendices to this document.

Acknowledgments

The following people have contributed to defining and documenting the SIERRA Requirements Management Process described in this document:

- Carter Edwards, Christi Forsythe, and Kathy Aragon of the Production Computing/ SIERRA Architecture Department
- Donna Eaton, Molly Ellis, and Kathleen Byle of the Information Technologies and Data Modeling Department
- Dwayne Knirk of the Quality Engineering Department
- Shelley Eaton of the Advanced Decision Support Applications Department
- Pam Harris of the Distributed Decision Support Systems Department

Contents

Acronyms and Abbreviations	7
1 Introduction	9
1.1 Purpose	9
1.2 Scope	9
2 Background	11
2.1 Definitions	11
2.2 Objective	12
2.3 Goals	12
2.3.1 Process Goals	12
2.3.2 Organizational Goal	13
2.4 Change Management Strategy	14
2.5 Layered Set of Integrated Requirements	14
2.5.1 Weapon Design Requirements (Layer 0)	15
2.5.2 Programmatic Requirements (Layer 1)	15
2.5.3 Physics and Functional Requirements (Layer 2)	15
2.5.4 Modeling and Simulation Requirements (Layer 3)	15
2.5.5 Software Requirements (Layer 4)	15
2.6 Requirements Topology Example	16
2.7 Software Process Improvement Approach	16
3 Requirements Management Process Overview	18
3.1 Purpose	18
3.2 Requirements Management Process Concepts	18
3.3 Roles and Responsibilities	20
3.3.1 Customer	20
3.3.2 Team Leader (TL)	20
3.3.3 Change Control Board (CCB)	20
3.3.4 Developer (DEV)	21
3.3.5 Quality Assurer (QA)	21
3.3.6 The Requirements Management Tool	21

4	Detailed Process Descriptions	22
4.1	Requirements Change Process	22
4.2	Requirements Topology Change Process	26
4.3	Requirements Attribute Change Process.....	28
4.4	Administrative Process	29
4.5	Requirements Attributes	29
5	Interfaces with Other Processes	31
5.1	Issue Tracking Process	31
	References	34

List of Figures

Figure 1.1	Hierarchical decomposition of the Requirements Domain (3 levels).....	10
Figure 2.1	Example Requirements Topology diagram.....	16
Figure 3.1	Interactions of the four Requirements Management processes diagram....	19
Figure 4.1	RCP (Requirements Change Process) diagram.....	24
Figure 4.2	RTCP (Requirements Topology Change Process) diagram.....	27
Figure 4.3	RACP (Requirements Attribute Change Process) diagram.....	28
Figure 4.4	AP (Administrative Process) diagram.....	29
Figure 5.1	Combined SCR Issue Tracking and Requirements Management Processes.	32

Acronyms and Abbreviations

AP	Administrative Process
ASCI	Accelerated Strategic Computing Initiative
CCB	Change Control Board
CMM	Capability Maturity Model
CMS	Configuration Management System
DB	database
DEV	Developer
KPA	Key Process Area
PMS	Project Management System
QA	Quality Assurer
RACP	Requirements Attribute Change Process
RCP	Requirements Change Process
RMS	Requirements Management System
RTCP	Requirements Topology Change Process
SCR	Software Change Request
SCM	Software Configuration Management
TL	Team Leader

Intentionally Left Blank

1 Introduction

1.1 Purpose

The purpose of this document is to provide a high-level description of the Requirements Management Process being developed as a SIERRA software process improvement task. The goal is to document the process for both managers and developers of the SIERRA infrastructure and the SIERRA applications. This documentation also meets the requirements of the software quality initiatives being implemented by the ASCI Program [1].

This document describes the *what* of requirements management and not the *how*. Furthermore, since the *how* is not being presented, no specific tool is recommended or discussed in detail. It is possible that there are several implementations of a requirements management process that would successfully carry out the process described in this document. Therefore, even if the implementation or the toolset changes in the future, the processes remain stable. An implementation of this requirements management process will be defined in *SIERRA Requirements Management Implementation with DOORS* [3].

Additional information about the SIERRA Requirements Management Process can be found in the following documents:

- *SIERRA Requirements Management Policy* [2]
- *SIERRA Project Dictionary* [8]

1.2 Scope

The SIERRA Requirements Management Process described in this document is required for each SIERRA software project that will develop delivered products. Each SIERRA project can extend the process to meet project-specific needs. The main sections of this document focus on the process requirements. Project-specific extensions should be included in appendices of this document.

Authors in the requirements discipline have used the terms “requirements engineering” and “requirements management” to refer to the whole discipline. In the book *Software Requirements* [4], Karl Wiegiers subdivides the requirements domain as shown in Figure 1.1. (The terminology in parentheses has been added to show the relationship between Wiegier’s terminology and the terminology used by other authors.)

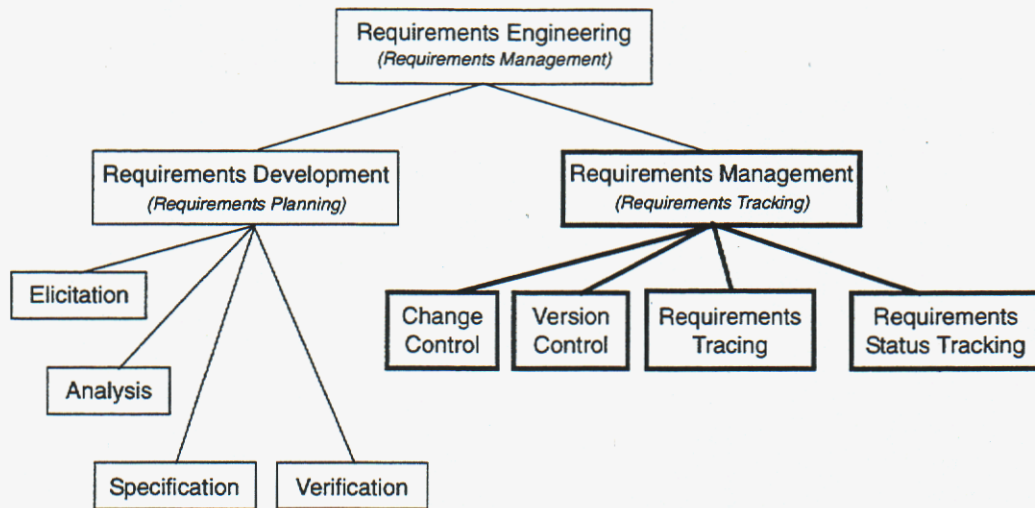


Figure 1.1. Hierarchical decomposition of the Requirements Domain (3 levels).

The requirements management process described in this document focuses on the “Requirements Management” activities shown **in bold** on the right side of the figure. The major activities in “Requirements Development” are not addressed in this process.

2 Background

2.1 Definitions

A *requirement*, as defined in Wiegers' *Software Requirements* [4] and the *Disciplined Requirements Process Seminar* course materials [10], is a statement or model identifying a capability, physical characteristic, or quality factor that bounds a product need for which a solution will be pursued. This requirement is needed to solve a problem or achieve an objective that must be met or possessed by a system or system component to satisfy formal or informal contractual obligations. Requirements address the following types of products:

- Delivered products;
- Enabling products to develop, support, maintain, and retire the delivered products;
- Interfaces with the intended environment; and
- Interfaces between products.

Requirements answer the questions:

- What is needed?
- What is not allowed?
- How well must a product perform? How fast, how often, how many, how much, how far, or how long?
- Under what conditions must a product perform?

Requirements Management entails establishing and maintaining an agreement with the customer on the requirements for the software project. That agreement is embodied in the written requirements specifications and models. It includes these areas:

- Baselineing of requirements;
- Analyzing proposed changes;
- Incorporating approved requirements;
- Tracing requirements from their source to designs, source code, and test cases; and
- Tracking status and change activities.

Additional definitions relevant to Requirements Management are provided in the *SIERRA Project Dictionary* [8].

2.2 Objective

The objective of Requirements Management for SIERRA is to establish a disciplined process whereby managers and developers of the SIERRA infrastructure and the SIERRA applications share a common understanding of both the requirements specifications and how to evaluate, approve, and communicate requirement changes. This process is critical to SIERRA's ability to produce the right products on schedule, within budget, and with high quality.

An obstacle to establishing a common understanding of the requirements definitions is the existence of several viewpoints or perspectives, all legitimate, based on the degree of detail and precision relevant to a particular role. For example, requirements are documented at a high level of generalization for a Sandia program manager, while a software developer demands more specificity and precision. An approach to overcoming this obstacle is to establish each viewpoint, organize the viewpoints into different layers based on the degree of generalization appropriate to the roles involved, and build traceability from the specific to the more generalized requirements definitions.

2.3 Goals

2.3.1 Process Goals

The goals of the SIERRA Requirements Management Process are as follows:

- Ensure that product requirements are well defined before incorporating them in the requirements repository, meaning that there are no conflicting requirements and that they are correct, necessary, clear, attainable, traceable, and verifiable;
- Maintain an evolving set of well-defined requirements;
- Establish an evolving structure for this set of requirements that clearly and concisely expresses the scope of requirements allocated to functional groups and requirements-derived interdependencies of those groups;
- Enable traceability from weapon program design support requirements through the integrated requirements into computing system capabilities;
- Identify ownership for each requirement and its implementation commitments;
- Maintain consistency among requirements and commitments when adding or changing requirements; and
- Integrate the Requirements Management Process with all other SIERRA software engineering processes.

2.3.2 Organizational Goal

A SIERRA organizational goal is to be prepared for a formal assessment of SIERRA for obtaining a Level 2 Capability Maturity Model [7] (CMM) rating. One of the Key Process Areas (KPA) in Level 2 of the CMM is Requirements Management. These KPAs map directly into the ASCI site practices [9]. Any Sandia formal assessment will measure against the site practices.

The purpose of the Requirements Management KPA is “to establish a common understanding between the customer and the software project of the customer’s requirements that will be addressed by the software project.”

Requirements Management involves establishing and maintaining an agreement with the customer on the requirements for the software project. The agreement covers both the technical and nontechnical requirements. An example of a nontechnical requirement is a delivery date. The agreement forms the basis for estimating, planning, performing, and tracking the software project’s activities throughout the software life cycle. Whenever the system requirements allocated to software are changed, the affected software plans, work products, and activities are adjusted to remain consistent with the updated requirements.

Goals of the CMM Requirements Management KPA include

1. Controlling system requirements allocated to software to establish a baseline for software engineering and management use. This is a baseline under Software Configuration Management (SCM). *Keeping track of what the requirements are and which ones are to be taken care of by the software.*
2. Keeping software plans, products, and activities consistent with the system requirements allocated to software (traceability).

The following CMM key practices, such as abilities, activities, and measurements, are important either because they are the activities that support a particular goal or because they have historically been a pitfall for organizations that have undergone assessments.

1. **Activity 1** – The software engineering group reviews its allocated requirements before they are incorporated into the software project. (This key practice is often a pitfall for organizations seeking to attain CMM Level 2 or 3).
2. **Activity 2** – The software engineering group uses its allocated requirements as the basis for software plans, work products, and activities.
3. **Activity 3** – Changes to the allocated requirements are reviewed and incorporated into the software project. (This key practice is often a pitfall for organizations seeking to attain CMM Level 2 or 3).

2.4 Change Management Strategy

The following change management strategies, documented in the *Introduction to a Disciplined Requirements Process Seminar* course materials [10], will be employed to facilitate the SIERRA Requirements Management Process:

- All requirements changes, regardless of origin, will follow the same process;
- All change proposals require justification and description that meet the evaluation criteria for well-defined requirements (correct, necessary, clear, attainable, traceable, and verifiable);
- The Change Control Board (CCB) will determine whether the proposed changes are necessary and worth the impact; and
- Work may not begin until the change is formally approved, except where a waiver is issued and implementation is immediate, as discussed in the *SIERRA Issue Tracking* document [5].

2.5 Layered Set of Integrated Requirements

The SIERRA software process improvement team generated a simple example illustration of how requirements flow down through the layers of a large program, such as the Department of Energy's Stockpile Stewardship program. In this simplified example, the requirements for SIERRA projects can be followed from the uppermost external source (Layer 0) through at least four layers of derived requirements:

- Uppermost External Source
 - Layer 0: Weapon Design Requirements
- Four Layers of Derived Requirements
 - Layer 1: Programmatic
 - Layer 2: Physics/Functional
 - Layer 3: Modeling/Simulation
 - Layer 4: Software

In general, the traceability between layers requires that for any *what* requirement in a particular layer, there must be some *why* requirements in the previous layer and some *how* requirements in the subsequent layer, assuming that these layers exist. From a SIERRA project's viewpoint, this means that for any *what* requirement for the SIERRA software (layer 4), there must be some *why* requirements in the modeling/simulation layer (layer 3) and some *how* requirements in the SIERRA software design.

2.5.1 Weapon Design Requirements (Layer 0)

Weapon Design Requirements come from sources outside of Sandia, such as DOD and DOE, but may also include some corporate-level requirements.

2.5.2 Programmatic Requirements (Layer 1)

The ASCI Programmatic Requirements are derived from analysis of Weapon Design Requirements, such as "Certify X will not catastrophically fail when ..." Programmatic Requirements address the design of (numerical) experiments and the simulation of objects in environments exhibiting behaviors. This level of requirements defines the questions to be answered by the simulation and matches ASCI capabilities to weapon design work, such as "Simulate X doing Y in environment Z."

2.5.3 Physics and Functional Requirements (Layer 2)

Physics and Functional Requirements are derived from analysis of Programmatic Requirements and address physical phenomena, physical characteristics, accuracy, and the analysis of experiments for experimental data needs. This layer of requirements defines the simulation capabilities to be used and provides simulation validation objectives. Physics and Functional Requirements match science to engineering, such as "Determine the range of maximum shear at this point."

2.5.4 Modeling and Simulation Requirements (Layer 3)

Modeling and Simulation Requirements are derived from analysis of Physics and Functional Requirements and address model equations and discretizations or algorithms. This layer of requirements defines the models to be simulated and provides model verification objectives. Modeling and Simulation Requirements represent science in models and solution methods, such as "Use this equation, those physical parameters, and that discretized volume."

2.5.5 Software Requirements (Layer 4)

Software Requirements are derived from analysis of Modeling and Simulation Requirements and address such things as software behavior, architecture, control and data structures, component interfaces, input/output, communications, and the computing environment. This layer of requirements trades off nonfunctional constraints such as available hardware and software and performance and timeliness. Software Requirements define the computational solutions to models and provide software verification objectives.

2.6 Requirements Topology Example

An example of a requirements topology that has been created to reflect SIERRA's current organizational structure is shown in Figure 2.1. A topology is a way of organizing and tracking requirement interrelationships. This topology organizes tightly coupled requirements into Requirements Management Sets, such as the SIERRA Framework and the SIERRA Tool Set. This ensures that strongly interdependent requirements remain together, as these requirements must be treated as a set in order to proceed forward.

It is possible that this topology will change. The Requirements Topology Change Process has been defined to manage this kind of change. However, the underlying layered set of integrated requirements shown in Figure 2.1 will remain stable.

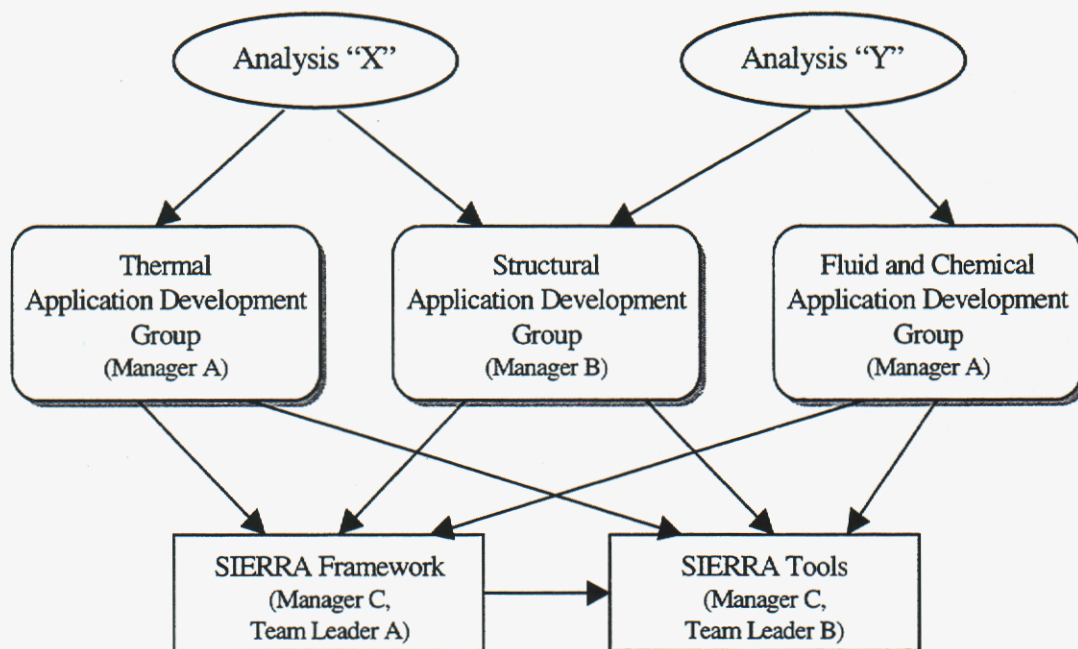


Figure 2.1. Example Requirements Topology diagram.

2.7 Software Process Improvement Approach

The model used for enhancing our capability to produce software is a usable, understandable, and disciplined engineering approach for continuous improvement. The model focuses on managing the improvement process and establishes the foundation for a long-term improvement strategy. There are several phases to this approach:

- Obtaining management and staff support to establish the framework for this effort;
- Assessing the current state of the software process;

- Creating a plan to achieve the desired level of quality;
- Implementing the plan; and
- Evaluating progress made and learning from those experiences.

As additional SIERRA tools and applications are included, the SIERRA Requirements Management Process will be reviewed, evaluated, changed, and improved as needed. The SIERRA Requirements Management document will serve as a template for these future projects.

3 Requirements Management Process Overview

3.1 Purpose

The purpose of the Requirements Management Process described in this document is to standardize the manner in which each SIERRA software project manages its requirements during the life span of a software project. The Requirements Management Process ensures all requirements changes are consistently documented, objectively evaluated, agreed upon, prioritized, communicated, and tracked to completion.

The process described in this document provides information and guidance to SIERRA personnel involved in the management of system and software requirements. This guidance is intended to satisfy the process activities described for requirements management in *The Capability Maturity Model: Guidelines for Improving the Software Process* [7] and thus, the *ASCI Site Practices* [9]. The Requirements Management Process contained in this document provides guidance to the practitioners so that each project can consistently follow the standardized basic practices, but it may also extend the process as is appropriate for the specific site and/or project in accordance with the *SIERRA Requirements Management Policy* [2].

3.2 Requirements Management Process Concepts

Four Requirements Management Process concepts are defined for SIERRA projects. More details on each type of process are presented in the next section.

1. Requirements Change Process (RCP) refers to changing the *content* of the requirements in the repository;
2. Requirements Topology Change Process (RTCP) addresses modifications to the *organization* of the requirements in the repository;
3. Requirements Attribute Change Process (RACP) refers to changing the requirement *status* or other *metadata* attributes; and
4. Administrative Process (AP) includes activities to *maintain the system*, such as creating reports, controlling access, and installing upgrades.

The process diagram in Figure 3.1 shows the flow and interaction of the four Requirements Management processes with the Requirements Management System (RMS) database and with the Configuration Management System (CMS) database. The interaction with the Project Management System (PMS) database will be implemented as a process enhancement at a later date.

The conventions used in Figure 3.1 and subsequent process diagrams are shown below.

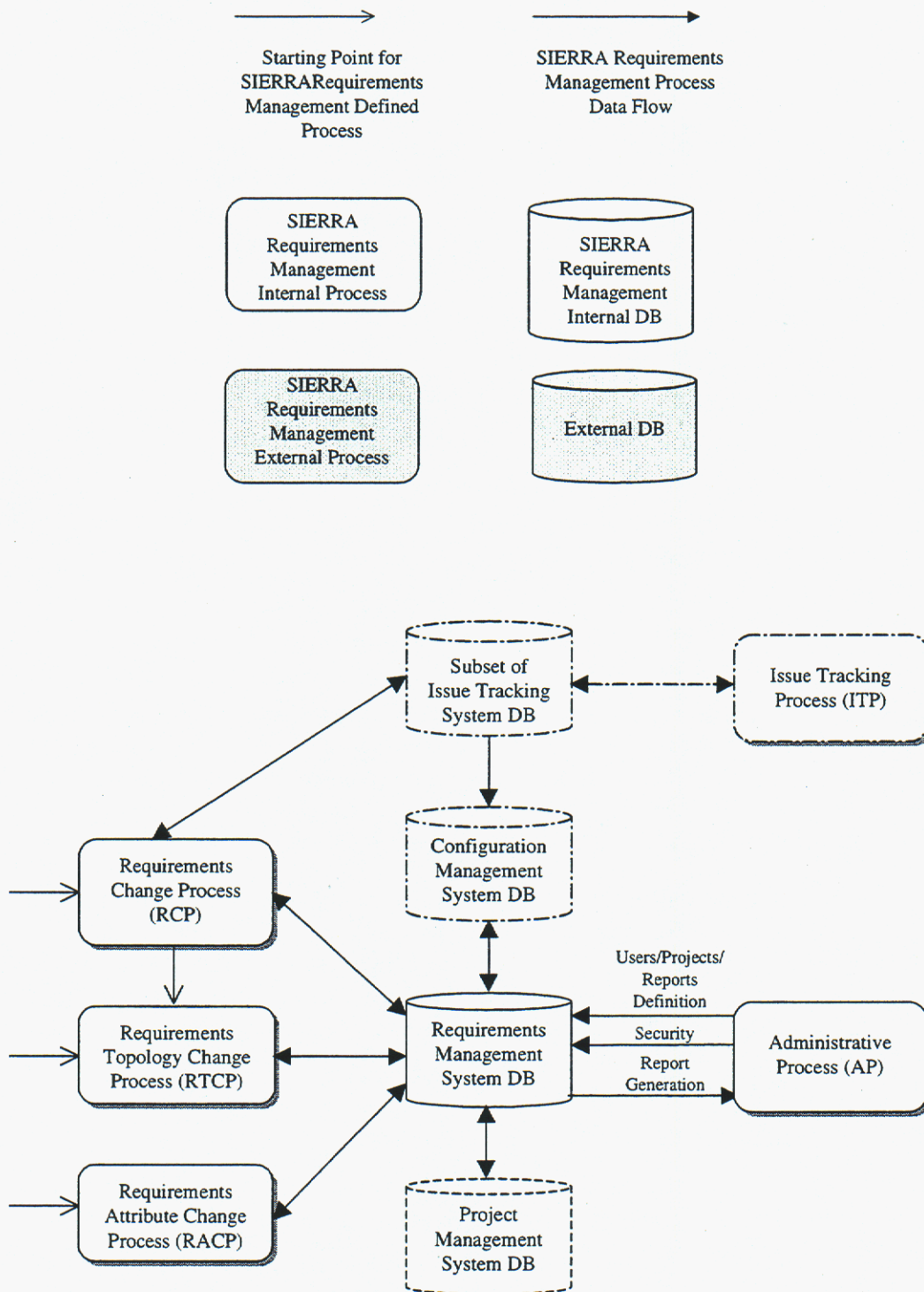


Figure 3.1. Interactions of the four Requirements Management processes diagram.

3.3 Roles and Responsibilities

A role is a part that a user plays in the SIERRA Requirements Management Process. A role may map to an individual or to a group, depending on the specific project. If more than one person plays a role, then one person from the group must be designated as being responsible for completing the tasks within the Requirements Management Process.

Whenever possible, the terms used for the roles in the SIERRA Requirements Management Process are consistent with the terms used for that individual or group in the SIERRA Issue Tracking Process. This document includes only the responsibilities defined for the roles relevant to the SIERRA Requirements Management Process.

3.3.1 Customer

The Customer is any person using a software application who wants to report a problem or request an enhancement. Customers submit *change requests* for enhancements or new capabilities.

The Customer's responsibility is to use the SIERRA Issue Tracking tool to report a problem or to request an enhancement. Customers who do not have access to the SIERRA Issue Tracking tool can request that a member of the software team submit a change request on their behalf.

3.3.2 Team Leader (TL)

A Team Leader is a member of the software team who is knowledgeable about the technical aspects of the team's product(s). Only one member of the team should play this role at any one time. This role may be rotated among the members of the team who have technical knowledge of the team's products. The Team Leader participates as a member of the Change Control Board.

The Team Leader is responsible for reviewing all customer requests submitted to the SIERRA Issue Tracking system, analyzing the requests, and developing a consistent Requirements Change Package for the technical requirements. This role coordinates the work on the requirements, along with tracking and recording progress. Upon approval of the requirements changes, the Team Leader updates and maintains these requirements changes in the Requirements Management Tool.

3.3.3 Change Control Board (CCB)

The Change Control Board is a group of people who review the Requirements Change Packages, which affect one or more software products, and negotiate the approval, deferral, or rejection of these packages.

The Change Control Board, which includes Team Leaders, is responsible for tracking progress on requirements and accepting the final implementation of requirements. The

Change Control Board periodically checks on requirements that have been deferred to see if any need to be re-opened.

3.3.4 Developer (DEV)

The Developer is a member of the software team who modifies documentation and source code to implement the solutions for requirements in the approved Requirements Change Packages. In a future enhancement to the Requirements Management Process, the Developer will also be responsible for updating the links between requirements and the software product modules that are modified to implement these requirements.

3.3.5 Quality Assurer (QA)

The Quality Assurer is a member of the software team who reviews the implemented changes to the team's products and verifies that all artifacts have been correctly modified. The Quality Assurer updates and maintains the release information (status and version) in the Requirements Management Tool. The Quality Assurer also reviews the activities of the software team to determine if the defined processes have been followed.

3.3.6 The Requirements Management Tool

The Requirements Management Tool is the requirements repository and interface for the requirements, Requirements Change Packages, attributes, and links. This tool is used to track any required history of these changes.

4 Detailed Process Descriptions

The general approach that is applied to the SIERRA Requirements Management Process is shown in the following list. This list includes the activities supported by this process, as specified in the previous section. More detailed information is then presented on each of the Requirements Management processes.

- Maintain a repository of requirements and linkages.
 - Define the structure of the repository.
 - Populate the repository with existing requirements, their attributes (metadata), and any linkages (optional).
 - Approve and baseline the repository.
- Receive requests for changes to the requirements repository.
- Select change requests to evaluate.
- Synthesize a consistent Requirements Change Package for the change request.
- Evaluate the Requirements Change Package for impact on current plans, activities, work products, and cost.
- Accept, reject, or defer the Requirements Change Package.
- Update the requirements repository, including metadata and any links, if appropriate.
- Establish the requirements baseline.
- Formally communicate approved changes to affected groups.
- Revise plans, activities, and work products to be consistent with the approved change.
- Track the requirements and requirement status in the Requirements Management Tool, including cost, schedule, and resources.
- Report status of requirements and change packages.

4.1 Requirements Change Process

The Requirements Change Process (RCP) encompasses the specification and integration of new requirements and modification or deletion of existing requirements. Integration, modification, or deletion of requirements includes associating or disassociating the requirement with one or more Requirements Management Sets. The topology organizes tightly coupled requirements into Requirements Management Sets. These requirements must be treated as a set in order to proceed forward.

Modification of requirements includes tracking the history of the modifications via baselines and version control. The capability to generate both standard and ad hoc reports is included in this process and will be provided as needs are identified.

The RCP (Requirements Change Process) is shown in Figure 4.1, using the conventions presented in Section 3.2. The RCP includes three analysis phases and ends with the decision phase:

- Request analysis, which is part of the Issue Tracking Process;
- Technical analysis, which is part of the Requirements Management Process;
- Resource analysis, which is part of a Project Planning Process; and
- Decision phase, which is part of the Requirements Management Process.

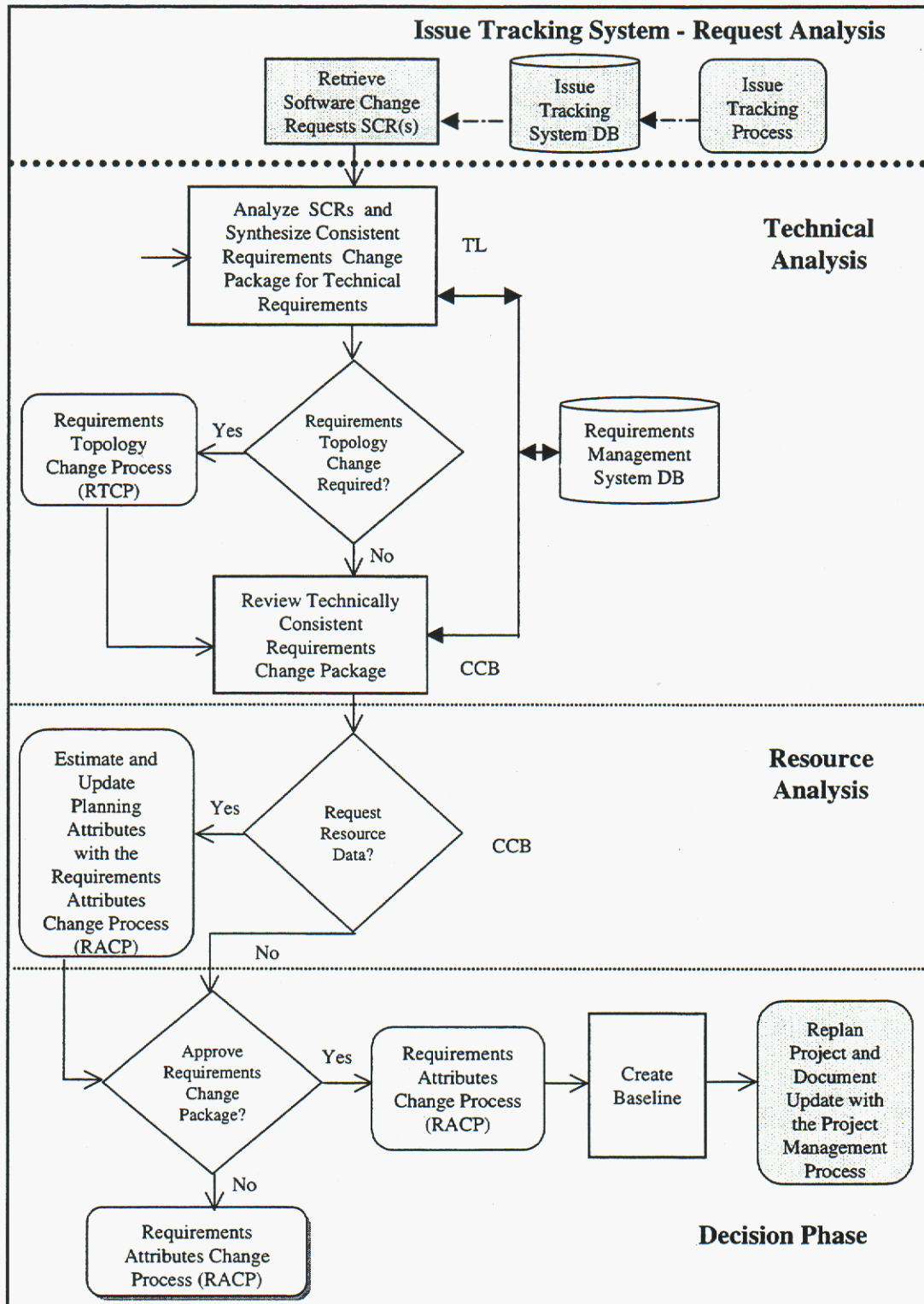


Figure 4.1. RCP (Requirements Change Process) diagram.

Request Analysis Phase

1. The Customer submits a Software Change Request (SCR) to the SIERRA Issue Tracking system. The current state of the request is *Submitted*. When an issue is logged, the state is updated to *Analyze*. The issue is analyzed to determine the disposition of the issue, which then determines what steps need to be taken to resolve it. This analysis could result in one or more requirements being generated or modified for the system.
2. The Requirements Management Process is invoked when the analysis of the SCR determines that the SCR's type has been appropriately assigned as an *enhancement request* or when the analysis of a *problem report* SCR determines that requirements need to be modified or added. The Team Leader creates a change proposal from the SCR and adds other known attributes to the User Requirements' Requirements Management Set.

Technical Analysis Phase

3. The requirements are organized into technically consistent Requirements Change Packages, providing for logical common groupings so that the appropriate teams review only relevant proposed requirements. This helps to mitigate the administrative burden.
4. The Team Leader completes the technical analysis of the Issue Tracking SCR(s) to synthesize the requirements. First, each SCR-derived requirement is allocated to a User Requirements' Requirements Management Set. Then the requirements that are derived from the user requirements are placed in other Requirements Management Sets, such as SIERRA Framework and the SIERRA Toolset. The Requirements Management System Database is updated and reports are generated for the review board. Each requirement will exist by association in only one Requirements Change Package at any given time. A Requirements Change Package contains one or more requirements, and these requirements are associated with one or more Requirements Management Sets. Any changes to a requirement's association with Requirement Management Sets or Requirements Change Packages is handled in this process.
5. If a Requirements Topology Change is required, the Requirements Topology Change Process (RTCP) is invoked by the Team Leader.
6. The Change Control Board (CCB) then reviews the technically consistent Requirements Change Package and/or the Requirements Topology Change request.
7. The Requirements Change Package, not individual requirements, is approved, rejected, or deferred. If only some of the requirements in a Requirements Change Package will be approved, the other requirements may be disassociated from the Requirements Change Package and placed into other Requirements Change Packages for later review so that the approved Requirements Change Package can move forward in the process.

8. The Requirements Management System Database is updated with all relevant information during the technical review.

Resource Analysis Phase

9. After the CCB reviews the Requirements Change Package, the CCB may request the Team Leader to perform a resource analysis before the CCB makes a decision on the change package. This resource analysis is linked to the Project Management Process.
10. If the CCB requests a resource analysis, the Team Leader estimates the project planning attributes for the Requirements Change Package. This estimation includes the cost and likelihood to implement the Requirements Change Package, a completion date, the likelihood of the completion date, the expected effect of accepting the Requirements Change Package on satisfying current commitments, and the rolling up of the schedule and costs.

Decision Phase

11. The CCB approves, rejects or defers the entire Requirements Change Package.
12. Regardless of the CCB's decision, the Team Leader then invokes the Requirements Attributes Change Process and updates the attributes of each requirement and each Requirements Change Package in the Requirements Management System Database.
13. If the Requirements Change Package is approved, requirements in the appropriate Requirements Management Set are updated and Requirements Change Packages will be updated in the Requirements Management Sets and baselined. The Requirements Management Set is baselined, and periodically the baselines will be published by producing a SAND report.
14. The CCB replans the project and documents the updated project plan. This is linked to the Project Management Process. Through the Issue Tracking Process, the Team Leader assigns implementation responsibility for new and changed requirements, and transitions the Requirements Change Package to the IT state of *Approve*. The CCB obtains appropriate organizational commitments.

4.2 Requirements Topology Change Process

The Requirements Topology Change Process (RTCP) encompasses all changes to the specification of Requirements Management Sets. A topology organizes tightly coupled requirements into the sets. The requirements must be treated as a set in order to proceed forward. The changes include definition of one or more new Requirements Management Sets (which are initially empty), updates to the attributes of one or more Requirements Management Sets, and updates to the relationships between Requirements Management Sets. The CCB owns all of the steps in the Requirements Topology Change Process (RTCP). The steps included in the RTCP are shown in Figure 4.2. An example that

represents the current situation is shown in Figure 2.1. The conventions used in this diagram are presented in Section 3.2,

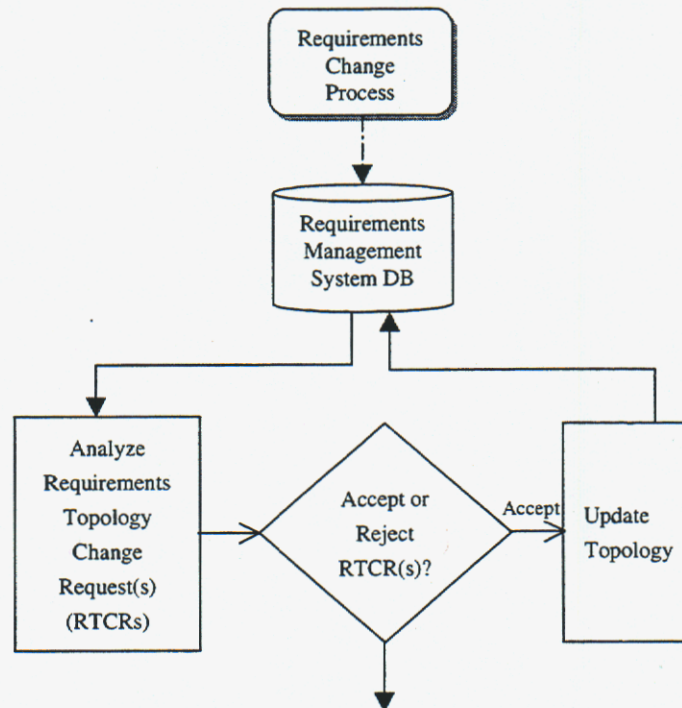


Figure 4.2. RTCP (Requirements Topology Change Process) diagram.

During the technical analysis of an SCR, a topology change may be needed. These steps are followed in that process.

1. The CCB analyzes the requirements topology change request(s) in the Requirements Change Package.
2. The CCB accepts, rejects, or defers the Requirements Change Package.
3. The Tool Administrator updates the topology, which alters the organizational structure of the repository. A topology organizes tightly coupled requirements into Requirements Management Sets. These requirements must be treated as a set in order to proceed forward.
4. A historical log of any structure changes will be kept.
5. Notification will be sent to the appropriate parties.

4.3 Requirements Attribute Change Process

The Requirements Attribute Change Process (RACP) modifies the metadata attributes of Requirements Change Packages, existing requirements, and Requirements Management Sets. After a requirement has been approved, the Status attribute is changed to *Approved*. Other examples of attributes include Estimated Effort, Reason for Rejection, and Target Product Release. The attributes that are changed are dependent on the process step.

The RACP does not modify the requirements topology, the association of requirements to Requirements Management Sets or Requirements Change Packages, or the content of any requirement. One or more roles can update one or more attributes. An individual's current role will define which attributes that individual should update for a specified activity.

This process includes updating the status attribute of requirements in the repository. For example, when code that satisfies a requirement is implemented and verified, the status of that requirement is updated accordingly. The capability to generate both standard and ad hoc reports is included in this process and will be provided as needs are identified.

The steps included in the RACP are shown in Figure 4.3, using the conventions presented in Section 3.2.

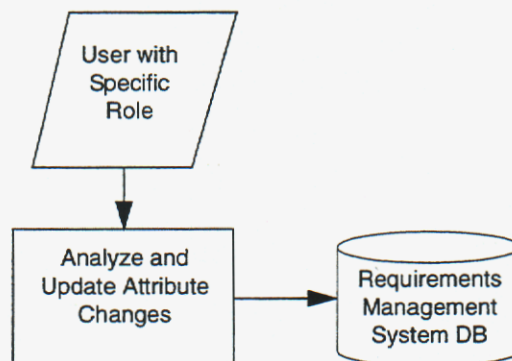


Figure 4.3. RACP (Requirements Attribute Change Process) diagram.

1. The user acting in a particular role (Team Leader, Quality Assurer, Change Control Board, etc.) identifies the attributes to be updated.
2. The Requirements Management Tool determines if the user has been given the appropriate privileges to change the attribute, and if allowed the user makes the change.
3. History is kept of all specified attribute changes.

4.4 Administrative Process

The Administrative Process (AP) includes report generation, assignment of each individual's roles, project initiation, and maintaining the system with backups and upgrades. The steps included in this process are shown in Figure 4.4, using the conventions presented in Section 3.2.

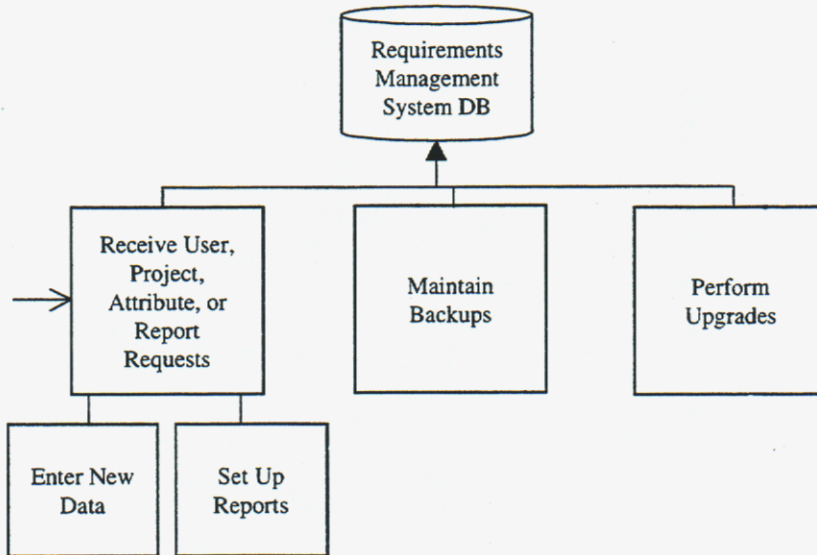


Figure 4.4. AP (Administrative Process) diagram.

1. The Tool Administrator receives the request for a adding, modifying, or deleting users, projects, attribute types, or reports. He then determines whether the request can be implemented and implements if possible.
2. The Tool Administrator ensures that the system administrator maintains backups for the system and the database and also performs tool upgrades.

4.5 Requirements Attributes

Requirements attributes that are managed by this process can be aggregated into the following categories:

- Identification and description
- Creation and modification history
- Customer request traceability
- Derived requirement traceability
- Implementation information

- Status
- Customer needs for validation and verification

5 Interfaces with Other Processes

5.1 Issue Tracking Process

The SIERRA Requirements Management Process works in conjunction with the established Issue Tracking Process. Requirements management and issue tracking are interwoven processes. The Requirements Management Process will use parts of the existing Issue Tracking Process, including the Submission, Notification, and Close activities.

An SCR within the Issue Tracking Process may be a problem report or request for enhancement. An enhancement request may have a very broad scope or be expressed in terms of a desired usage. An issue will rarely be specified as a well-defined and consistent requirement. It is the responsibility of the Team Leader to synthesize requirements from issues. Subsequent processing of the synthesized requirements must be reflected in the originating issues.

Figure 5.1 shows the high-level connections between these two processes. The process flow on the left is the SCR Issue Tracking, with the Requirements Management flow on the right.

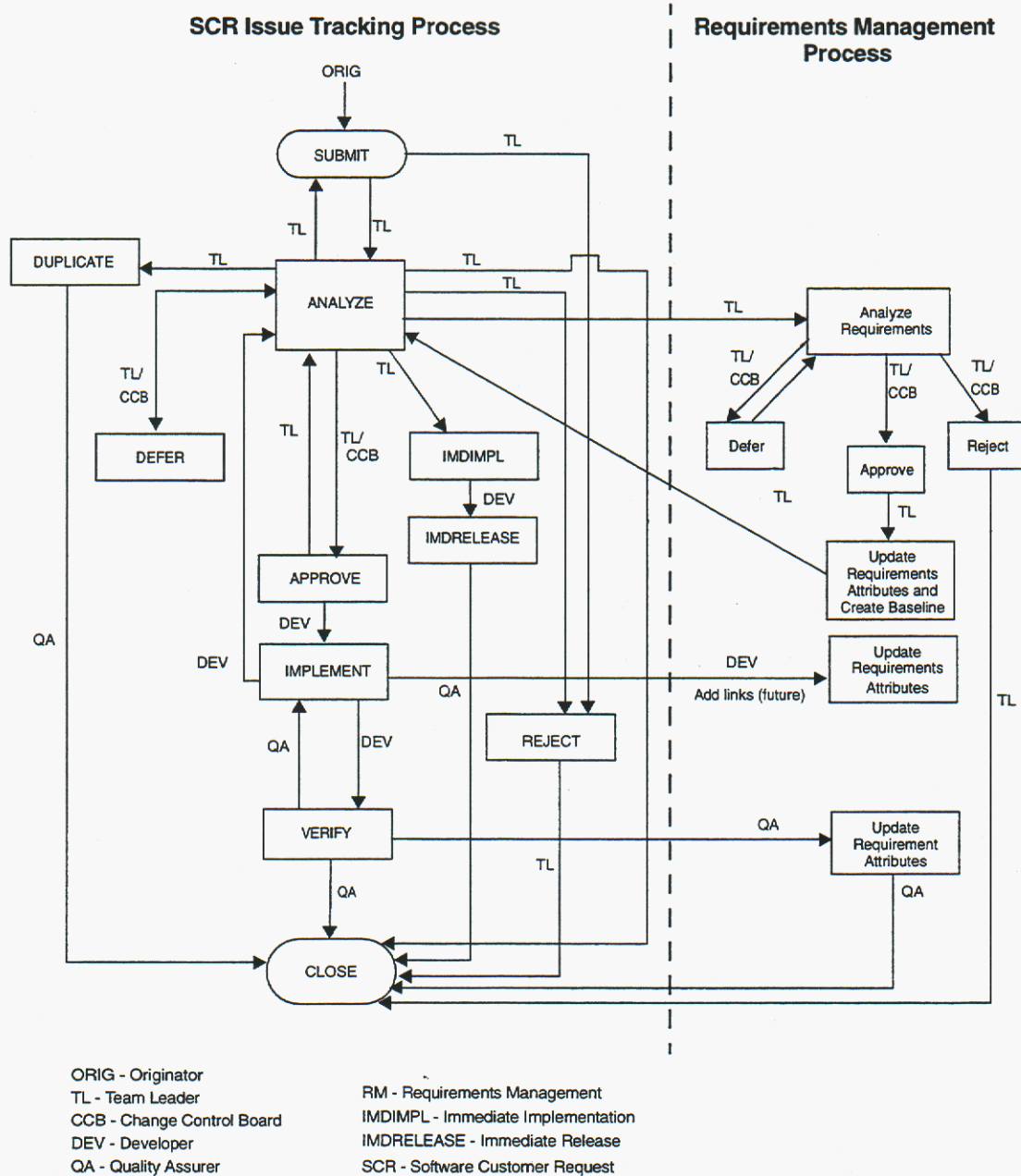


Figure 5.1. Combined SCR Issue Tracking and Requirements Management Processes.

There are two sources for requirements for the SIERRA software products:

- Synthesized directly from Customer requests and
- Derived from the analysis of higher-layer requirements or requirements that are too coarse-grained to guide planning and development.

The SIERRA Issue Tracking Process [5] provides the Customer interface to the SIERRA Requirements Management Process as well as the tracking mechanism for the implementation and verification of Customer requests. In the Issue Tracking Process, the SCR process model is used when a customer requests a modification.

The Requirements Management Process is invoked initially when the analysis of a submitted SCR determines that the SCR's type has been appropriately assigned as an *enhancement request* or when the analysis of a *problem report* SCR determines that requirements need to be modified or added.

In a future enhancement of the Issue Tracking and Requirements Management Processes, the Requirements Management Process may be invoked during the implementation activity of the Issue Tracking Process for the Developer to update the requirements' linking attributes.

The Requirements Management Process is invoked from the Issue Tracking Process during verification so that the Quality Assurer can update the requirement attributes.

References

1. Ann Hodges, Gary Froelich, David Peercy, Martin Pilch, and Juan Meza (Sandia National Laboratories), Maysa-Maria Peterson, Jeanette LaGrange, Lawrence Cox, and Ken Koch (Los Alamos National Laboratory), and Nancy Storch, Cynthia Nitta, and Evi Dube (Lawrence Livermore National Laboratory), *ASCI Software Quality Engineering: Goals, Principles, and Guidelines*, DOE/DP/ASC-SQE-2000-FDRFT-VERS2, February 2001.
2. *Policy*, SIERRA Requirements Management, DRAFT, Albuquerque, NM: Sandia National Laboratories-Engineering Sciences Center, June 2000.
3. *SIERRA Requirements Management Implementation with DOORS*, DRAFT, Albuquerque, NM: Sandia National Laboratories-Engineering Sciences Center, April 2001.
4. Karl E. Wiegers, *Software Requirements*. Redmond, Washington: Microsoft Press, 1999.
5. *Issue Tracking*, SIERRA Tools Configuration Management, August 2000.
6. Lee M. Taylor, Harold Carter Edwards, and James R. Stewart, *Functional Requirements for SIERRA Version 1.0 Beta*, SAND99-2587, Albuquerque, NM: Sandia National Laboratories-Engineering Sciences Center, October 1999.
7. Mark C. Paulk, Charles V. Weber, Bill Curtis, and Mary Beth Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process*, 1994.
8. *Project Dictionary*, SIERRA Project, Albuquerque, NM: Sandia National Laboratories-Engineering Sciences Center, February 2001.
9. Ann Hodges, Gary Froehlich, David Peercy, and Martin Pilch, *Sandia National Laboratories ASCI Software Quality Engineering: Practices and Work Products*, 2/14/2001, DRAFT, Version 1.0, Albuquerque, NM: Sandia National Laboratories-Engineering Sciences Center.
10. *Introduction to a Disciplined Requirements Process Seminar*, Software Productivity Consortium, 2001.

Distribution

1	MS9003	D. L. Crawford, 9900
1	MS 0139	J. E. Kelly, 9904
1	MS 0841	T. C. Bickel, 9100
1	MS 0833	M. A. Robertson, 9103
1	MS 0824	A. C. Ratzel, 9110
1	MS 0826	W. L. Hermina, 9113
1	MS 0834	J. E. Johannes, 9114
1	MS 0825	W. H. Rutledge, 9115
1	MS 0836	E. S. Hertel, 9116
1	MS 0836	R. O. Griffith, 9117
1	MS 0847	H. S. Morgan, 9120
1	MS 0555	M. S. Garrett, 9122
1	MS 0847	D. R. Martinez, 9124
1	MS 0557	T. J. Baca, 9125
1	MS 0847	R. A. May, 9126
1	MS 0824	J. L. Moya, 9130
1	MS 0828	M. Pilch, 9133
1	MS 1135	S. R. Heffelfinger, 9143
1	MS 0827	J. M. McGlaun, 9140
1	MS 0828	J. A. Fernandez, 9140
1	MS 0835	S. N. Kempka, 9141
1	MS 0835	J. S. Peery, 9142
5	MS 0827	J. D. Zepper, 9143
10	MS 0827	K. M. Aragon, 9143
20	MS 0827	H. C. Edwards, 9143
1	MS 0827	C. A. Forsythe, 9143
1	MS 0827	G. D. Sjaardema, 9143
1	MS 0827	J. R. Stewart, 9143
1	MS 0827	K. D. Copps, 9143
1	MS 0827	A. B. Williams, 8920
1	MS 1137	S. M. Eaton, 6536
1	MS 1137	P. W. Harris, 6536
1	MS 1137	M. Tebo, 6536
1	MS 1137	A. Hodges, 6536
1	MS 1137	L. Claussen, 6536
1	MS 1137	S. Moore, 6536
1	MS 1137	D. Moore, 6536
1	MS 1137	B. Spicer, 6536
1	MS 1137	M. Williamson, 6536

1	MS 1137	G. Froehlich, 6536
1	MS 1137	H. Ogden, 6536
1	MS 1137	S. Romero, 6536
1	MS 1138	M. McCornack, 6536
1	MS 0660	J. A. Larson, 9519
1	MS 0660	D. S. Eaton, 9519
1	MS 0660	M. E. Ellis, 9519
1	MS 0638	K. A. Byle, 9519
1	MS 0638	D. L. Knirk, 12326
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0612	Review and Approval Desk, 9612
		For DOE/OSTI